



FUNDAMENTOS DE SQL



FUNDAMENTOS DE SQL

- Abordagem relacional e utilização dos operadores relacionais
- Diferenças entre os sub-conjuntos de comandos SQL
- Comando SELECT
- Sub consultas simples e correlacionadas, e combinação com os operadores do SQL
- Comandos INSERT, UPDATE, DELETE enquadrados em transacções
- Criação de tabelas e utilização de restrições
- Criação de índices simples
- SQL como linguagem “universal” para pesquisas sobre bases de dados
- Pesquisas (“queries”) simples sobre a base de dados (estrutura básica do comando SELECT)
- Predicados ALL e DISTINCT
- Pesquisas complexas
- Agregação de dados com a instrução SELECT
- Lógica e funções de grupo
- JOIN como forma de extrair informação de tabelas diferentes com base em critérios de comparação de valores em colunas comuns (INNER JOIN, LEFT JOIN e RIGHT JOIN)
- Utilização de sub pesquisas (ou pesquisas encadeadas)
- Uniões



*Ana Marta Nunes Grade
DBA administrator*

Experiencia Académica:
Licenciada em Engenharia Informática
Pós-graduação em Gestão de Projectos
Pós-graduação em WebDesign

Experiencia Profissional:
Força Aérea Portuguesa
BNP Paribas
Banco Investimento Global - BIG



Nome?

Função actual?

Experiência Académica?

Experiência Profissional?

Perspectivas com esta formação?

ABORDAGEM RELACIONAL E UTILIZAÇÃO DOS OPERADORES RELACIONAIS

PROGRAMAÇÃO EM LINGUAGEM SQL

As Bases de Dados foram desenvolvidas para:

- Fornecer acesso facilitado aos dados (DATA);
- Gestão de grandes volumes de informação (DATA);
- Responder a todo o tipo de perguntas ao sistema;
- Relacionar-se com as mais variadas bases dados existentes e tabelas existentes.

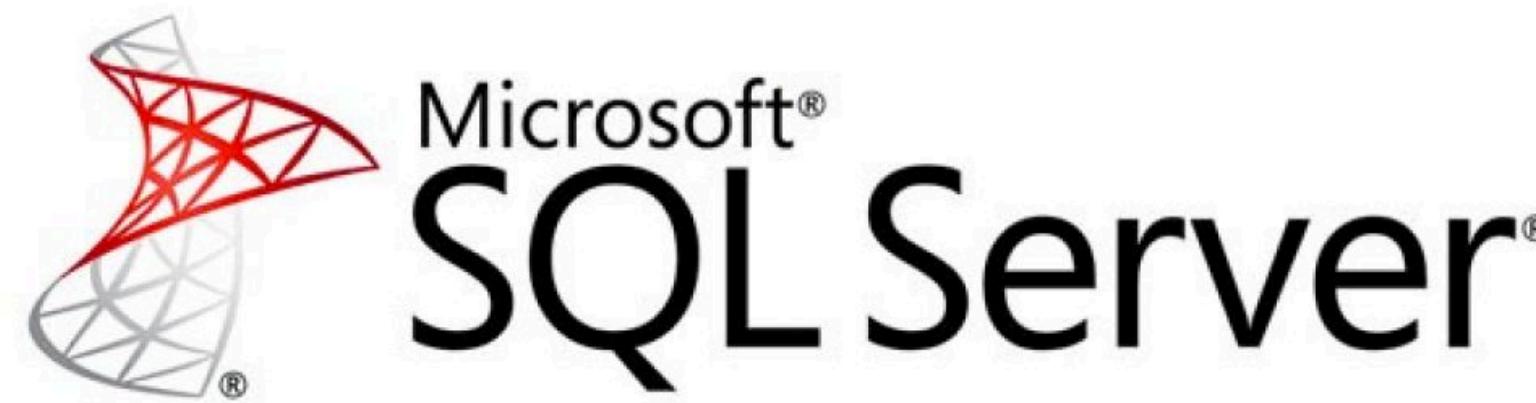
A linguagem padrão dos Bancos de Dados Relacionais é a **Structured Query Language**, ou simplesmente **SQL**, como é mais conhecida.

PROGRAMAÇÃO EM LINGUAGEM SQL

As bases dados são:

- Conjuntos de dados com uma estrutura regular que organizam informação;
- Um conjunto de informações relacionadas entre si, referentes a um mesmo assunto e organizadas de maneira útil, com o propósito de servir de base para que o utilizador recupere informações, tire conclusões e tome decisões.
- (Database) Aplicativos de software cujo objectivo é compilar, organizar e armazenar informações em meio electrónico de forma estruturada.
- Conjunto de informações armazenadas e ordenadas para consulta imediata por meio de um palavra-chave (Primary key)

PROGRAMAÇÃO EM LINGUAGEM SQL - EXEMPLOS



ORACLE



MODELO RELACIONAL

PROGRAMAÇÃO EM LINGUAGEM SQL - MODELO RELACIONAL

Uma bases de Dados segue uma abordagem relacional, ou seja insere-se num Modelo Relacional.

O que é um Modelo Relacional?

O **Modelo Relacional** é um modelo de dados, adequado a ser o modelo subjacente de um **Sistema Gerenciador de Banco de Dados (SGBD)**, que se baseia no princípio de que todos os dados estão armazenados em tabelas e os relacionamentos inerentes às mesmas.

PROGRAMAÇÃO EM LINGUAGEM SQL - MODELO RELACIONAL

O que é um Sistema Gerenciador de Banco de Dados (SGBD)?

- O SGBD é responsável por tudo, armazenar os dados no HD;
- Manter em memória os dados mais acedidos;
- Ligar dados e metadados;
- Disponibilizar uma interface para programas e utilizadores externos que accessem o banco de dados (para banco de dados relacionais, é utilizada a linguagem SQL);
- Encriptação de dados;
- Controlar o acesso a informações;
- Manter cópias dos dados para recuperação de uma possível falha;
- Garantir transações no banco de dados.

PROGRAMAÇÃO EM LINGUAGEM SQL - MODELO RELACIONAL



PROGRAMAÇÃO EM LINGUAGEM SQL - MODELO RELACIONAL

A arquitetura de uma Base de Dados pode ser descrita informalmente e formalmente:

- Na descrição informal - utilização de termos mais práticos como tabela, linha e coluna.
 - Na descrição formal - utilização de termos mais técnicos como relação (tabela), tupla(linhas) e atributo(coluna).

Relação -> Tabela

PROGRAMAÇÃO EM LINGUAGEM SQL - MODELO RELACIONAL

Tabelas - Estrutura de linhas e colunas;

Registros ou Tuplas - Contém informações ou até mesmo poderão assumir valores nulos;

Colunas ou Atributos - As colunas podem ser denominadas por atributos uma vez que temos que identificar um atributo (tipo de dados).

Chaves - As tabelas relacionam-se entre si através de chaves:

Chaves Primárias (PK) - Trata-se de um Id que identifica univocamente um registo (dado);

Chaves Estrangeiras (FK) - é a chave formada através de um relacionamento com a chave primária de outra tabela

RELACIONAMENTOS

PROGRAMAÇÃO EM LINGUAGEM SQL - RELACIONAMENTOS

Existem alguns tipos de relacionamentos possíveis no Modelo de Entidades e Relacionamentos:

- Um para um (**1 para 1**) - indica que as tabelas têm relação unívoca entre si.
- Um para muitos (**1 para N**) - a chave primária da tabela que tem o lado 1 “desloca-se” para a tabela do lado N.
- Muitos para muitos (**N para N**) - quando tabelas têm entre si relação n..n, é necessário criar uma nova tabela com as chaves primárias das tabelas envolvidas, ficando assim uma chave composta, ou seja, formada por diversos campos-chave de outras tabelas. A relação então se reduz para uma relação 1..n, sendo que o lado n ficará com a nova tabela criada.

NORMALIZAÇÃO

PROGRAMAÇÃO EM LINGUAGEM SQL - NORMALIZAÇÃO

A normalização de dados é um processo que simplifica grupos complexos de dados para evitar redundâncias e possibilitar um maior desempenho nas pesquisas.^[1]

É o processo de organização eficiente dos dados dentro de uma base de dados cujos objetivos principais são:

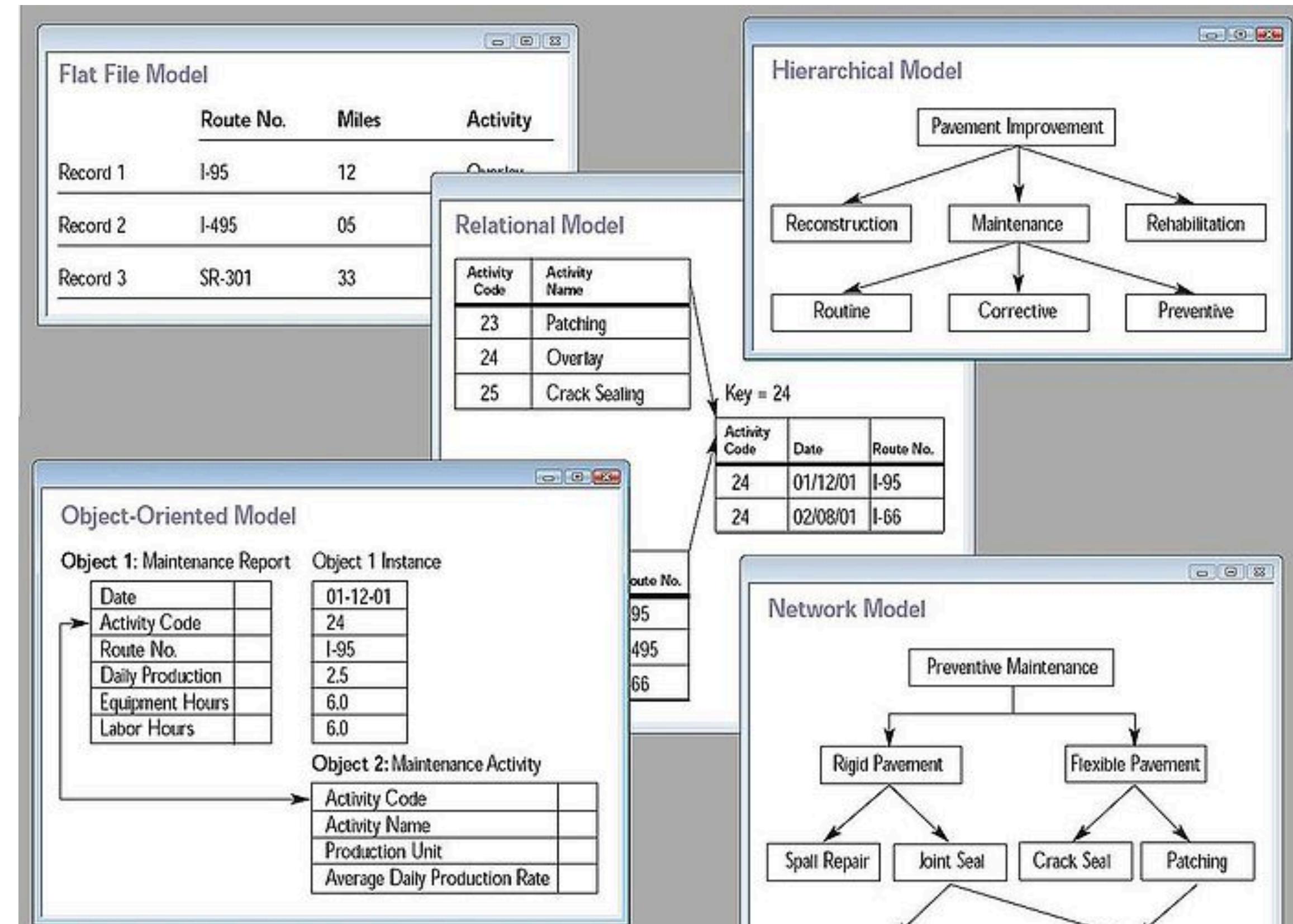
1. Eliminar dados redundantes (por exemplo, armazenando os mesmos dados em mais de uma tabela).
2. Garantir que as dependências entre os dados façam sentido (armazenando apenas dados logicamente relacionados em uma tabela).

MODELOS DE BASES DADOS

PROGRAMAÇÃO EM LINGUAGEM SQL - MODELOS

Existem vários Modelos de Base de Dados:

- Modelo Plano (ou tabular);
- Modelo em Rede;
- Modelo Hierárquico;
- Modelo Relacional;
- Orientado a objetos;
- Objeto-Relacional.



PROGRAMAÇÃO EM LINGUAGEM SQL - MODELOS

- O **modelo plano** consiste de matrizes simples, bidimensionais, compostas por elementos de dados: inteiros, números reais, etc.;
- O **modelo em rede** permite que várias tabelas sejam usadas simultaneamente por meio do uso de apontadores (ou referências). Algumas colunas contêm apontadores para outras tabelas ao invés de dados. Assim, as tabelas são ligadas por referências, o que pode ser visto como uma rede;
- O **modelo hierárquico** é uma variação particular do modelo em rede, limita as relações a uma estrutura semelhante à de uma árvore (hierarquia - tronco, galhos), ao invés do modelo mais geral direcionado por grafos;

PROGRAMAÇÃO EM LINGUAGEM SQL - MODELOS

Bases de dados relacionais consistem, principalmente de três componentes:

- Uma coleção de estruturas de dados, conhecidas como relações;
- Uma coleção dos operadores (cálculo relacionais);
- Uma coleção de restrições da integridade, definindo o conjunto consistente de estados de base de dados e de alterações de estados. As restrições de integridade podem ser de quatro tipos: de domínio (também conhecidas como *type*), de atributo, variável relacional e restrições de base de dados.

TRANSAÇÕES ACID

PROGRAMAÇÃO EM LINGUAGEM SQL - TRANSAÇÕES ACID

Uma transação é um conjunto de procedimentos, executados num banco de dados em que o utilizador percebe como uma única ação.

A integridade de uma transação depende de **quatro propriedades**, conhecidas como **ACID**:

- **Atomicidade:**

Todas as ações que compõem a unidade de trabalho da transação devem ser concluídas com sucesso, para que seja efetivada (**commit**).

Se durante a transação qualquer ação que constitui unidade de trabalho falhar, a transação inteira deve ser desfeita (**rollback**).

Quando todas as ações são efetuadas com sucesso, a transação pode ser efetivada(**commit**).

PROGRAMAÇÃO EM LINGUAGEM SQL - TRANSAÇÕES ACID

- **Consistência**

Todas as regras e restrições definidas no banco de dados devem ser utilizadas. Relacionamentos por chaves estrangeiras, verificação de valores para campos restritos ou únicos devem ser obedecidos para que uma transação possa ser completada com sucesso.

- **Isolamento**

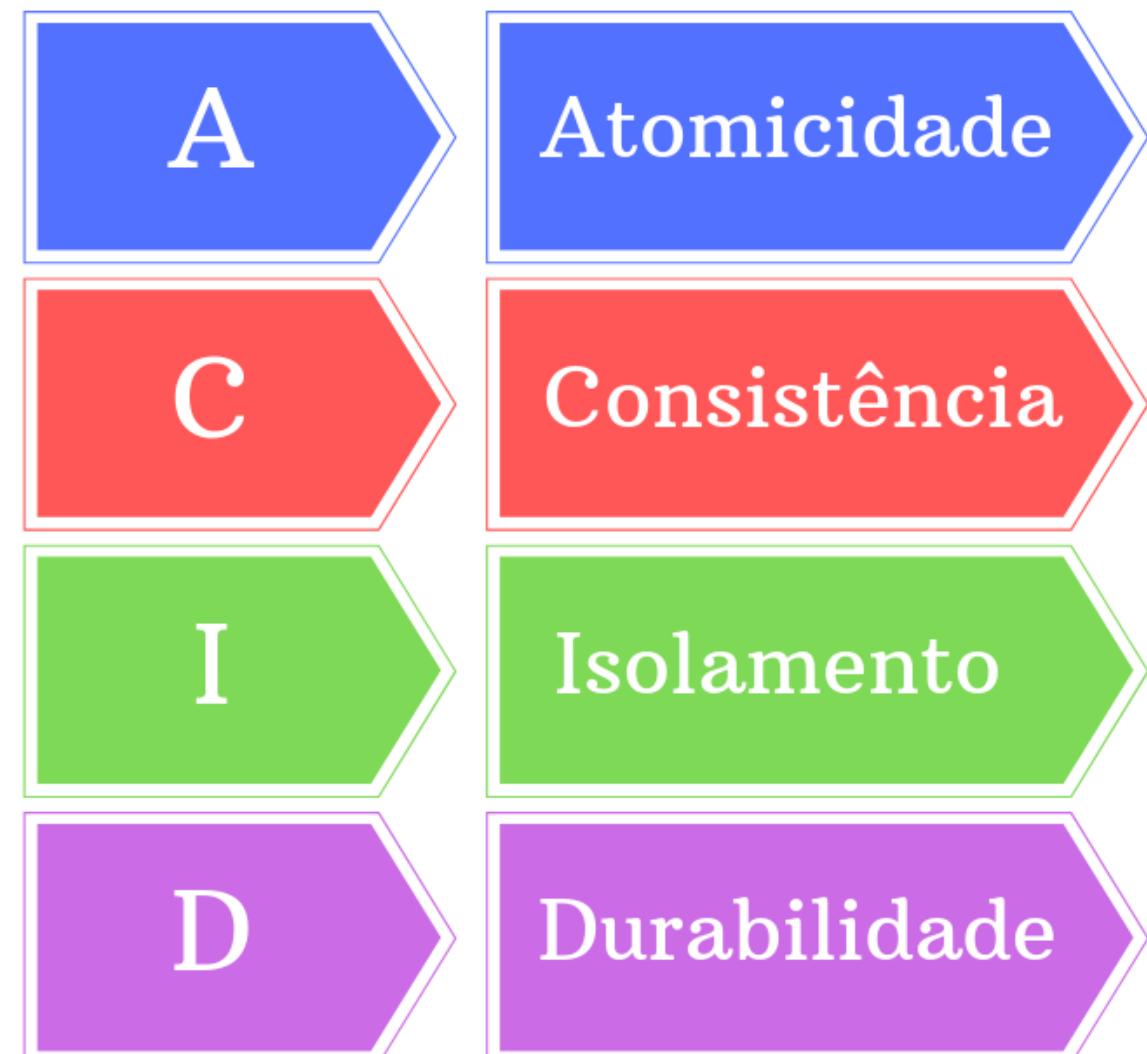
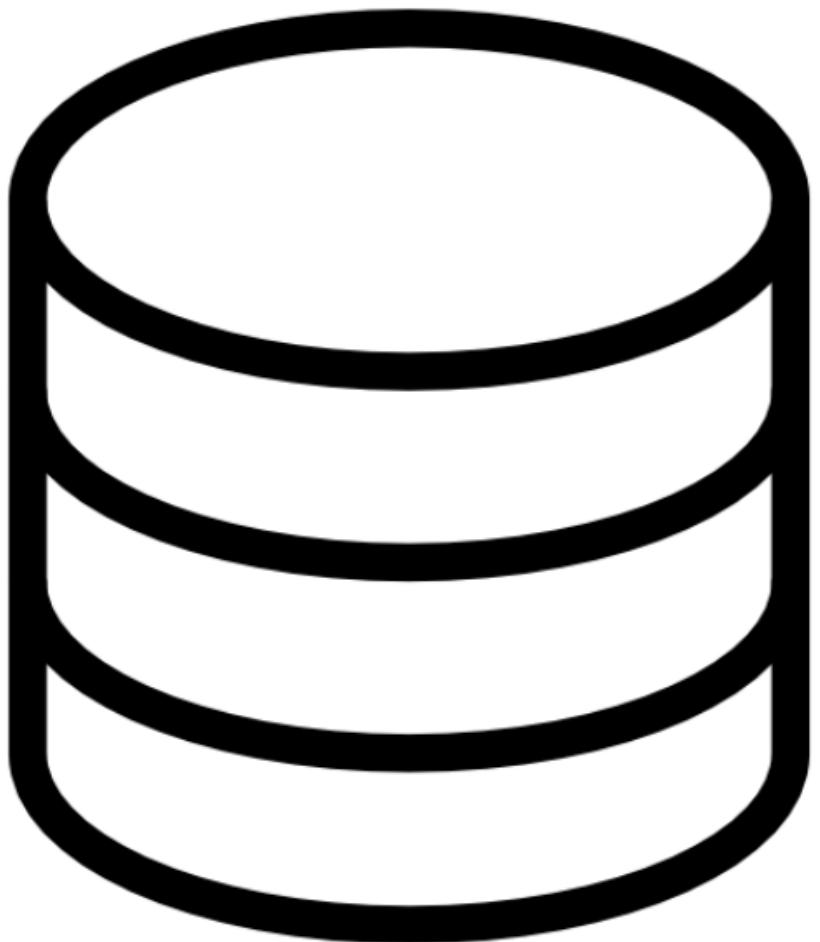
Cada transação funciona completamente à parte de outras transações. Todas as operações são parte de uma transação única. O princípio é que nenhuma outra transação, operando no mesmo sistema, possa interferir no funcionamento da transação corrente (trata-se de um mecanismo de controlo).

PROGRAMAÇÃO EM LINGUAGEM SQL - TRANSAÇÕES ACID

- **Durabilidade**

Significa que os resultados de uma transação são permanentes e podem ser desfeitos somente por uma transação subsequente.

Por exemplo: todos os dados e status relativos a uma transação devem ser armazenados num repositório permanente, não sendo passíveis de um falha de hardware.

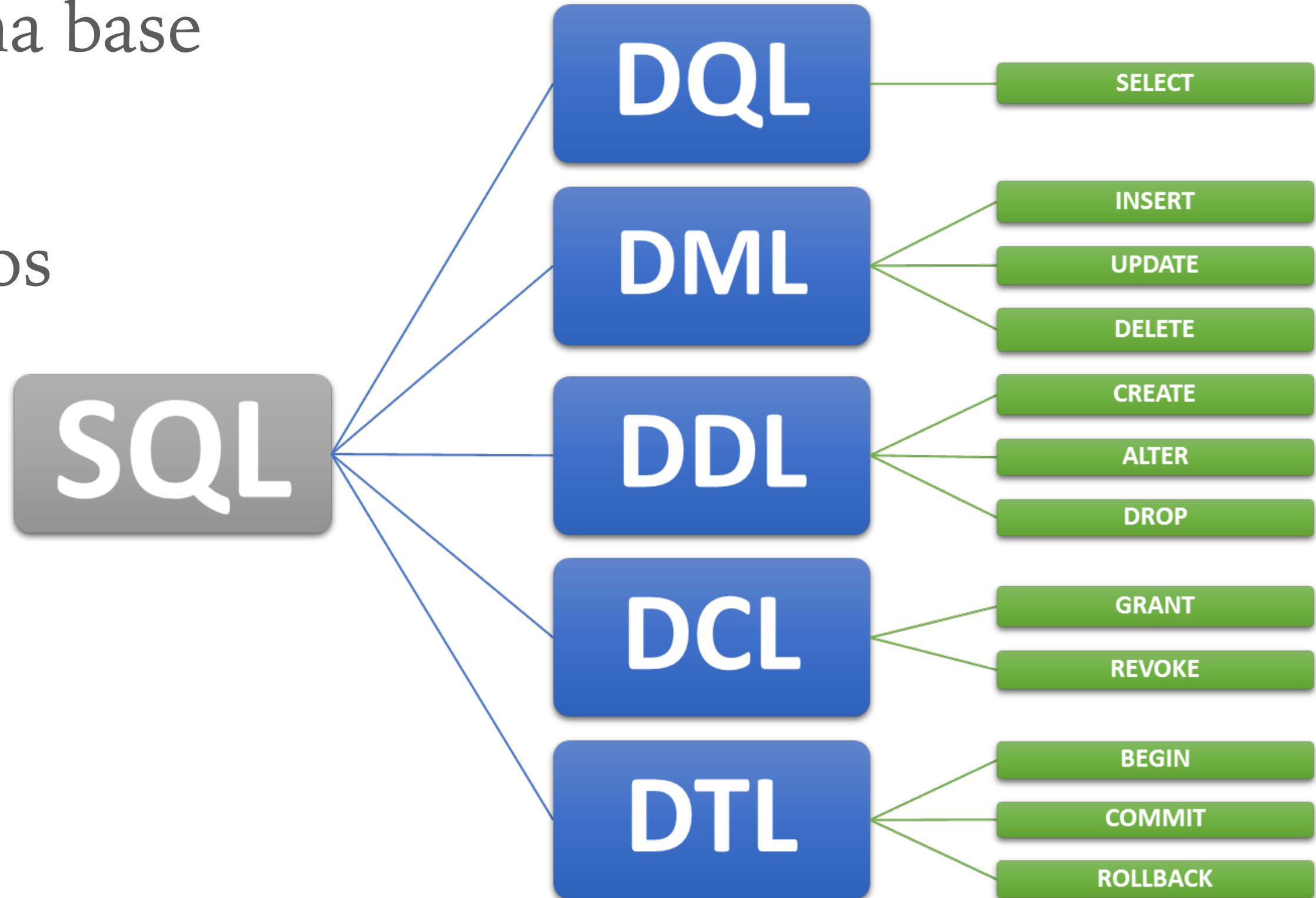


SUBCONJUNTOS SQL

PROGRAMAÇÃO EM LINGUAGEM SQL - SUBCONJUNTOS SQL

A linguagem SQL é dividida em subconjuntos de acordo com as operações que queremos efetuar sobre uma base de dados, tais como:

- DML - Linguagem de Manipulação de Dados
- DDL - Linguagem de Definição de Dados
- DCL - Linguagem de Controle de Dados
- DTL - Linguagem de Transação de Dados
- DQL - Linguagem de Consulta de Dados



PROGRAMAÇÃO EM LINGUAGEM SQL - SUBCONJUNTOS SQL

DML - Linguagem de Manipulação de Dados

O primeiro grupo é a **DML** (Data Manipulation Language - Linguagem de manipulação de dados).

DML é um subconjunto da linguagem SQL que é utilizado para realizar inclusões, consultas, alterações e exclusões de dados presentes em registos.

Estas tarefas podem ser executadas em vários registos de diversas tabelas ao mesmo tempo. Os comandos que realizam respectivamente as funções acima referidas são **Insert**, **Update** e **Delete**.

PROGRAMAÇÃO EM LINGUAGEM SQL - SUBCONJUNTOS SQL

FUNÇÃO	COMANDOS SQL	Descrição do Comando	EXEMPLO
Inserção de registos	INSERT	Usada para inserir um registo a uma tabela existente.	INSERT INTO Pessoa (id, nome, sexo) VALUE;
Alterações aos registos	UPDATE	Mudar os valores de dados em uma ou mais linhas da tabela existente.	UPDATE Pessoa SET data_nascimento = '11/09/1985' WHERE id_pessoa = 7
Remover registos	DELETE	Permite remover linhas existentes de uma tabela.	DELETE FROM pessoa WHERE id_pessoa = 7

PROGRAMAÇÃO EM LINGUAGEM SQL - SUBCONJUNTOS SQL

DDL - Linguagem de Definição de Dados

DDL (Data Definition Language - Linguagem de Definição de Dados).

Uma DDL permite ao utilizador definir tabelas novas e elementos associados.

Os comandos básicos da DDL são poucos:

- **CREATE**: cria um objeto (uma **Tabela**, por exemplo) dentro da base de dados.
- **DROP**: apaga um objeto da base de dados.

Alguns sistemas de bases de dados usam o comando **ALTER**, que permite ao usuário alterar um objeto, por exemplo, adicionando uma coluna a uma tabela existente.

PROGRAMAÇÃO EM LINGUAGEM SQL - SUBCONJUNTOS SQL

Outros comandos *DDL*:

- CREATE TABLE - Criar Tabela
- CREATE INDEX - Índice
- CREATE VIEW - View
- ALTER TABLE - Alterar Tabela
- ALTER INDEX - Alterar Índice
- DROP INDEX - Remover Índice
- DROP VIEW - Remover View

PROGRAMAÇÃO EM LINGUAGEM SQL - SUBCONJUNTOS SQL

DCL - Linguagem de Controle de Dados

DCL (Data Control Language - Linguagem de Controle de Dados).

DCL controla os aspectos de autorização de dados e licenças de utilizadores para controlar quem tem acesso para ver ou manipular dados dentro do banco de dados.

Duas palavras-chaves da DCL:

- **GRANT** - autoriza os utilizadores executar ou configurar operações.
- **REVOKE** - remove ou restringe a capacidade de um utilizador de executar operações.

PROGRAMAÇÃO EM LINGUAGEM SQL - SUBCONJUNTOS SQL

DTL - Linguagem de Transação de Dados

- **BEGIN** - (ou **BEGIN TRANSACTION**, dependendo da sintaxe de SQL) - pode ser usado para marcar o começo de uma transação de banco de dados que pode ser completa ou não.
- **COMMIT** - finaliza uma transação dentro de um sistema de gestão de base de dados.
- **ROLLBACK** - faz com que as mudanças nos dados existentes desde o último **COMMIT** ou **ROLLBACK** sejam descartadas.

COMMIT e **ROLLBACK** interagem com áreas de controlo como transação e alocação. Ambos terminam qualquer transação aberta e liberam qualquer cadeado ligado a dados.

PROGRAMAÇÃO EM LINGUAGEM SQL - SUBCONJUNTOS SQL

DQL - Linguagem de Consulta de Dados

Embora tenha apenas um comando, a DQL é a parte da SQL mais utilizada.

O comando **SELECT** permite ao utilizador especificar uma consulta ("query") a fim de obter o resultado pretendido. Esse comando é composto de várias cláusulas e opções, possibilitando elaborar consultas das mais simples às mais elaboradas.

FUNÇÃO	COMANDOS SQL	Descrição do Comando	EXEMPLO
Consultas	<u>SELECT</u>	O Select é o principal comando usado em SQL para realizar consultas a dados pertencentes a uma tabela.	Select * From Pessoa;

CLÁUSULAS SQL

PROGRAMAÇÃO EM LINGUAGEM SQL - CLÁUSULAS SQL

CLÁUSULA	DESCRICAÇÃO	QUERY
FROM	Utilizada para especificar a tabela, que se vai selecionar os registos.	Select * from TabelaA
WHERE	Utilizada para especificar as condições que devem reunir os registos que serão selecionados.	Select * from TabelaA where id = 1
GROUP BY	Utilizada para agrupar os registos selecionados em grupos específicos.	Select * from TabelaA group by name
HAVING	Utilizada para expressar a condição que deve satisfazer cada grupo.	Select * from TabelaA having(idade) > 21
ORDER BY	Utilizada para ordenar os registos selecionados com uma ordem específica.	Select * from TabelaA order by name
DISTINCT	Utilizada para selecionar dados sem repetição.	Select distinct * from TabelaA
UNION	Combina os resultados de duas consultas SQL em uma única tabela para todas as linhas correspondentes.	Select * from TabelaA union Select * from TabelaB

OPERADORES LÓGICOS

PROGRAMAÇÃO EM LINGUAGEM SQL - OPERADORES LÓGICOS SQL

CLÁUSULA	DESCRÍÇÃO	QUERY
AND	“E” lógico. Avalia as condições e devolve um valor verdadeiro caso ambos sejam corretos.	Select * from TabelaA where id = 1 AND name = ‘Ana’
OR	“OU” lógico. Avalia as condições e devolve um valor verdadeiro se algum for correto.	Select * from TabelaA where id = 1 OR name = ‘Ana’
NOT	Negação lógica. Devolve o valor contrário da expressão.	Select * from TabelaA where id is NOT null

OPERADORES RELACIONAIS

PROGRAMAÇÃO EM LINGUAGEM SQL - OPERADORES RELACIONAIS SQL

CLÁUSULA	DESCRÍÇÃO	QUERY	EXEMPLOS
<	Menor que	SELECT * FROM informacao.tabela WHERE idade < 18;	Seleciona todos os registos na "tabela" que possuem o campo "idade" com valores menores que 18.
>	Maior que	SELECT * FROM informacao.tabela WHERE idade > 18;	Seleciona todos os registos na "tabela" que possuem o campo "idade" com valores maiores que 18.
<=	Menor ou igual que	SELECT * FROM informacao.tabela WHERE idade <= 18;	Seleciona todos os registos na "tabela" que possuem o campo "idade" com valores menores ou iguais a 18.
>=	Maior ou igual que	SELECT * FROM informacao.tabela WHERE idade >= 18;	Seleciona todos os registos na "tabela" que possuem o campo "idade" com valores maiores ou iguais a 18.
(=)	Igual a	SELECT * FROM informacao.tabela WHERE idade = 18;	Seleciona todos os registos na "tabela" que possuem o campo "idade" com valores exatamente iguais à 18.
<> / !=	Diferente de	SELECT * FROM informacao.tabela WHERE idade <> 18;	Seleciona todos os registos na "tabela" que possuem o campo "idade" com valores que são diferentes de 18.

FUNÇÕES DE AGREGAÇÃO

PROGRAMAÇÃO EM LINGUAGEM SQL - FUNÇÕES DE AGREGAÇÃO SQL

CLÁUSULA	Descrição	QUERY	EXEMPLOS
AVG	Média	SELECT AVG(preco) FROM produto	Retorna a média do preço dentro da tabela Produto
COUNT	Contabilização	SELECT COUNT(id) FROM produto	Retorna a contagem de todos os id's presentes na tabela Produto
SUM	Soma	SELECT SUM(valor) FROM produto	Retorna a soma da coluna "valor" dentro da tabela Produtos
MAX	Máximo	SELECT MAX(preco) FROM produto	Retorna o valor máximo da coluna Preço da tabela produto
MIN	Mínimo	SELECT MIN(preco) FROM TabelaProduto	Retorna o valor mínimo da coluna Preço da tabela produto